MULTIPROCESSOR SYSTEM-ON-CHIP IMPLEMENTATION FOR NETWORK-ON-CHIP COMMUNICATION WITH 5-BIT DATA LINE

M.V.V.PRASANThI1, V.SREEvANTI2
1PG student (M.Tech), GIET, Rajahmundry, India
2Assistant Professor, ECE dept., GIET, Rajahmundry, India
1prasanthi.santhi.90@gmail.com, 2sreevanivaddi@gmail.com

ABSTRACT
This paper introduces the silicon-proven method of different On-Chip interconnection networks or Network-on-Chips (NoCs). These Network-on-Chips (NoCs) are becoming the de-facto scaling communication techniques in Multi-Processor System-on-Chip (MPSoC) or Chip Multiprocessor (CMP) environment. The prospected network design defined with a 5-bit data line. Under a multistage network topology, the proposed network consists of circuit-switching technique with a dynamic path-setup scheme. The dynamic path-setup scheme implements runtime path arrangement for arbitrary traffic permutations. The circuit-switching approach offers an assurance of permuted data and its compact overhead enables the benefit of stacking multiple networks.

Keywords: Assured output, multistage interconnection network, Network-on-Chip, network arrangement, pipelined circuit-switching, traffic permutation, switch, crossbar.

[1] INTRODUCTION
System-on-chip (SOC) design gets increasingly complex, as a growing number of applications are integrated in modern systems. Some of these applications have real-time requirements, such as a minimum throughput or a maximum latency. To reduce cost, system resources are shared between applications, making their timing behaviour inter-dependent. Real-time requirements must hence be verified for all possible combinations of concurrently executing applications, which is not feasible with commonly used simulation-based techniques. Applications in a composable system are completely isolated and cannot affect each other’s behaviours, enabling them to be independently verified. Predictable systems, on the other hand, provide lower bounds on performance, allowing applications to be verified using formal performance analysis. Five techniques to achieve predictability in SOC resources are presented and we explain their implementation for processors, interconnect. The complexity of contemporary Systems-on-Chip (SOC) is increasing, as a growing number of independent applications are integrated and executed on a single chip.

These applications consist of communicating tasks mapped on heterogeneous multi-processor platforms with distributed memory hierarchies that strike a good balance between performance, cost, power consumption and flexibility. The platforms exploit an increasing amount of application-level parallelism by enabling concurrent execution of more and more applications. This results in a large number of use-cases, which are different combinations of concurrently running applications. Some applications have real-time requirements, such as a minimum throughput of video frames per second, or a maximum latency for processing those video frames. Applications with real-time requirements are referred to as real-time applications, while the rest are non-real-time applications. A use-case can contain an arbitrary mix of real-time and non-real-time applications. This becomes a great challenge when these permutation networks need to be implemented under very limited on-chip power and area overhead.

Reviewing on-chip permutation networks (supporting either full or partial permutation) with regard to their implementation shows that most the networks employ a packet-switching mechanism to deal with the conflict of permuted data. Their implementations either use first-input first-output (FIFO) queues for the conflicting data, or time-slot allocation in the overall system with the cost of more routing stages, or a complex routing with a deflection technique that avoids buffering of the conflicting data. The choices of network design factors, i.e., topology, switching technique and the routing algorithm, have different impacts on the on-chip implementation. The topologies used here are direct and regular topologies like mesh and torus which are automatically possible for the physical design in a 2-D chip. Another issue in systems on-chip is traffic to and from shared resources. Shared resources typically attract an over proportional amount of traffic from many cores of the system or are the source of traffic toward all these cores. Examples include shared on-chip memories, transactional memory, hardware accelerators, and IOs. For certain accelerator cores or IOs, a possible option to optimize the communication cost of traffic from distant cores would be to instantiate multiple instances distributed over the whole system. However, this solution is not applicable for shared memories that, by definition, cannot be partitioned or distributed. For IOs, this approach might also be prohibitive in most cases due to pin count limitations of the chip package. To optimize the communication performance and decrease communication cost,
the communication infrastructure should be optimized for traffic to and from the shared resources. By reducing the network distance between the communication partners, not only the performance but also the latency and energy consumption can be decreased.

Different routing algorithms, buffer size allocation, and switch arbitration policies. Hu and Marculescu examined power-efficient mapping of a heterogeneous 16-core task graph representing a multimedia. This paper introduces a new silicon-proven design of an on-chip permutation network to sustain assured throughput of permuted traffics under random permutation. Unlike conventional packet-switching approaches, proposed on chip network communication results a circuit-switching mechanism with a dynamic path-setup scheme under a multistage network algorithm. The dynamic path setup manages the threat of runtime path arrangement for consistent permuted data. The fixed data paths enable a throughput guarantee. By removing the extreme overhead of queuing buffers, a compact implementation is attained and heaping multiple networks to hold parallel permutations in runtime is reasonable.

Finally, another important publication focuses on application traffic. Communication weighted application models consider communication aspects (CWM), while communication dependence and computation models (CDCM) simultaneously consider both application aspects.

2. PROPOSED ON-CHIP NETWORK DESIGN
As propelled in part I, the crucial thought of anticipated on-chip system outline is focused around a pipelined circuit-switching methodology with an element way setup plan carrying runtime way course of action. Before saying the element way setups conspire, the system topology is initially talked about. At that point the plans of diverting nodes are exhibited.

2.1 Network-topology
In Clos system, a group of multistage systems are connected to fabricate adaptable business multi-processors with a huge number of joints in large-scale frameworks. The run of the mill three-stage Clos system is characterized as C (n, m, p), where n speaks to the quantity of inputs in each of p first-arrange switches and m is the quantity of second stage switches. The end goal is to attain an equidistance level of 25 as in most commonsense MPSoCs. The prospected design uses C (5, 5, 5) as a topology for the planned system. This system has a rearrange able property that can understand all conceivable changes in its include and yields. The decision of the three-stage Clos system with a humble number of centre stage switches is to reduce accomplishment cost, whereas it still enables are arrange able behaviour of the network.

A pipelined circuit-exchanging plan is intended for utilization with the prospected system. The plan has the stages like the setup, the transfer, and the release. An element way setup plan supporting the runtime way course of action happens in the setup stage. To implement circuit-switching, the connection of two adjacent switches with its handshake signs is prospected as shown in fig2.

![Fig.1. On-Chip network topology](image1)

![Fig.2. Connection of two adjacent switches with its handshake signs](image2)
The bit arrangement of the handshake incorporates a 1-bit Request (Req) and a 2-bit Answer (Ans). Req = 1 is utilized when a switch asks for an moving connection prompting the comparing next switch in the setup stage. Req = 1 is additionally continued amid information exchange along the setup way. Req = 0 signified that the switch discharges the involved connection. This convention is likewise utilized as a part of both the setup and the discharge stages. If Ans = 01 implies that the end of the line is prepared to get information from the source. At the point when the Ans = 01 spreads again to the source, it indicates the way is to be found up, then an information exchange can be begin quickly. If Ans = 11 is saved for link-to-link stream control when the accepting circuit is not prepared to get information because of being occupied with different assignments, or flood at the getting cushion, and so on. If Ans = 10(back) implies that the connection is obstructed. This Back convention is utilized for a backpressure stream control of the element way setup plan, which is talked about in the accompanying subsection.

2.2 Dynamic Path Setup to Support Path Arrangement

An element way setup plan is the crucial purpose of the prospected outline to help a runtime way course of action when the stage is changed. Every way setup, which begins from information to discover a way prompting the terminus. The idea of testing is presented in works, in which a test (or setup flutter) is rapidly sent under a directing calculation so as to build a way towards the terminus. Depleted gainful backtracking (EPB) is prospected to use to course the test in the system work. The way course of action with full stage comprises of 25 ways to setup path, while a setup path with a fractional change may comprise of a subset of 25 way setups. An inquiry is that can the proposed EPB-based way setups utilized with the Clos C (5, 5, 5) understand all conceivable full stages between its inputs and yields? As sealed in works, the three-stage Clos system C (5, 5, 5) is rearrange able if m ≥ n. In the proposed system of C (5, 5, 5), m = n = 5, so it is rearrange able. There dependably exists an accessible way from a unmoving info prompting an unmoving yield. By the Exhaustive Property of EPB as sealed in work [12], the EPB-based way setup totally looks all the conceivable ways inside the set of way differing qualities between an unmoving include and unmoving yield. Specifically applying the Exhaustive Property of the inquiry into rearrange able C (5, 5, 5) demonstrates that the EPB-based way setup can simply find an accessible way inside the set of four conceivable ways between the information and the unmoving yield. Taking into account this EPB-based way setup plan, it is clear that the way course of action for full (and incomplete) stage can simply be acknowledged in the proposed system with C (5, 5, 5) topology.

As outlined in this system, each one data sends a test containing a 5-bit yield location to discover an accessible way prompting the target yield. Amid the inquiry, the test moves advances when it discovers a free connection and moves regressively when it confronts a blocked connection. By method for non-dreary development, the test finds an accessible way between the info and its comparing unmoving yield. The EPB-based way setup plan is planned with a situated of test directing calculations as said later. The accompanying illustration portrays how the way setup attempts to discover an accessible way by utilizing the set of way assorted qualities. It is accepted that a test from a source (e.g., a data of switch 01) is attempting to setup a way to a target end of the line (e.g.,
an accessible yield of switch 02). In the first place, the test will non-drearily attempt ways as the second progressed stage switches in the request of 10→11→12→13.

Expecting that the connection 01→10 is accessible, the test first tries this connection (Req =1) and afterward touches base at switch 22. If joint 10 - 22 is accessible, the test lands at switch 22 and meets the target yield. If Ans = Ack then engenders again to the data to trigger the exchange stage. If joint 10 - 22 is hindered, the test will move once again to switch 01 and join 01 - 10 is discharged. From switch 01, the test can then attempt whatever remains of unmoving connections prompting the second-organize switches in the same way. By method for moving back when confronting blocked connections and attempting others, the test can rapidly set up the way in runtime in a conflict-avoidance manner.

2.3 Switching Node Designs

Three sorts of switches are intended for the proposed on-chip system. These switches are all focused around regular switch building design, with the main distinction being in the test directing calculations. This basic construction modelling has essential segments: INPUT Controls (ICs), OUTPUT Controls (OCs), an ARBITER, and a CROSSBAR. Approaching tests in the setup stage can be transported through the information ways to save cost on wiring expenses.

The ARBITER has 2 capacities: in the first place, cross-joining the Ans_out and the ICs through the Grant transport, and second, as a ref for the appeals from the ICs. At the point when an approaching test lands at info, the relating IC watches the yield status through the Status transport, and appeals the ARBITER to give it get to to the comparing OC through the Request transport. At the point when tolerating this task for, the ARBITER cross-interfaces the relating Ans_out with the IC through the Grant transport with its first capacity. With the second capacity, the ARBITER, in view of a predefined need tenet, purpose conflict when a few ICs demand the same free yield. After this determination, one IC is acknowledged, though the rest are replied as confronting a blocked connection.

3. SIMULATION RESULTS
4. DESIGN SUMMARY
The performance simulations are carried out with Modelsim 6.3 for Debugging and Xilinx 14.3 for Synthesis. The test frequency is 154.416MHz.

<table>
<thead>
<tr>
<th>Logic Utilization</th>
<th>Used</th>
<th>Available</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slices</td>
<td>744</td>
<td>4656</td>
<td>15%</td>
</tr>
<tr>
<td>Number of Slice Flip Flops</td>
<td>365</td>
<td>9312</td>
<td>3%</td>
</tr>
<tr>
<td>Number of 4 input LUTs</td>
<td>1286</td>
<td>9312</td>
<td>13%</td>
</tr>
<tr>
<td>Number of bonded IOBs</td>
<td>269</td>
<td>66</td>
<td>407%</td>
</tr>
<tr>
<td>Number of GCLKs</td>
<td>2</td>
<td>24</td>
<td>8%</td>
</tr>
</tbody>
</table>

Table 1: Performance analysis

CONCLUSIONS
An on-chip network design aiding traffic permutations in MPSoC applications has been defined in this paper. Using the Clos network topology, the prospected design offers random transfer arrangements in runtime with solid implementation cost by using a circuit-switching technique together with dynamic path-setup scheme. The silicon-proven test-chip verifies the prospected design and recommends for use as on-chip infrastructure-IP aiding transfer arrangements in future MPSoC researchers.

REFERENCES